academia global

Universidad Corporativa Coppel | Programas Académicos

Ingeniería en Desarrollo de Software

Actividad | 1 | Estructuras de Control

Lenguajes de Programación IV

Ingeniería en Desarrollo de Software

TUTOR: Aarón Iván Salazar Macías

ALUMNO: Adriana Esteban López

FECHA: 28 de Julio de 2025

INDICE

INTRODUCCIÓN

Una estructura de control es aquella que nos indica de que manera deben de irse ejecutando las acciones que plasmamos en un diagrama de flujo, es decir, es el código que utilizamos dentro de las plataformas de programación.

Existen tres tipos de estructuras de control:

1. Secuenciales, estas son las más simples ya que las instrucciones se ejecutan una detrás de otra.
2. Condicionales, aquí ya se incorporan instrucciones y/o indicaciones que permiten al usuario seleccionar una u otra opción/acción (condiciones dobles o múltiples).
3. Iterativas, son aquellas que van a estar repitiendo una o varias instrucciones varias veces, ejemplo de estas serían **for, while, do-while**

El uso de las estructuras de control son indispensables para el éxito de programas funcionales; dentro del desarrollo de esta actividad estaremos haciendo uso tanto de estructuras secuenciales como condicionales.

Las secuenciales serán aquellas que ejecutamos en un orden ya que primero ingresamos el peso y posteriormente la estatura, con lo cual damos paso al calculo del IMC y después indicar en que clasificación esta.

Las estructuras condicionales se estarán utilizando a través del uso de If, else; ya que se estará realizando varias comparaciones con los datos ingresados para determinar la clasificación de la persona de acuerdo a su IMC.

DESCRIPCIÓN

Contextualización:

El IMC es el índice de masa corporal que cada persona tiene, lo cual se refiere a la masa y talla de la misma, para su cálculo existe una fórmula establecida, a su vez, existe una tabla la cual determina la clasificación de IMC que una persona tiene.

En México, se tiene uno de los más altos índices de masa corporal en su población, por lo que un hospital de la ciudad de México, necesita que se cree un programa que los ayude a calcular el IMC de sus pacientes.

Actividad:

Crear un sistema que calcule el IMC de los pacientes del hospital de la ciudad de México, haciendo uso del lenguaje de programación Java 8 y el entorno de programación sugerido en la sección de Recursos para realizar un programa con los siguientes requerimientos:

1. **Interfaz Datos que deberá solicitar:**

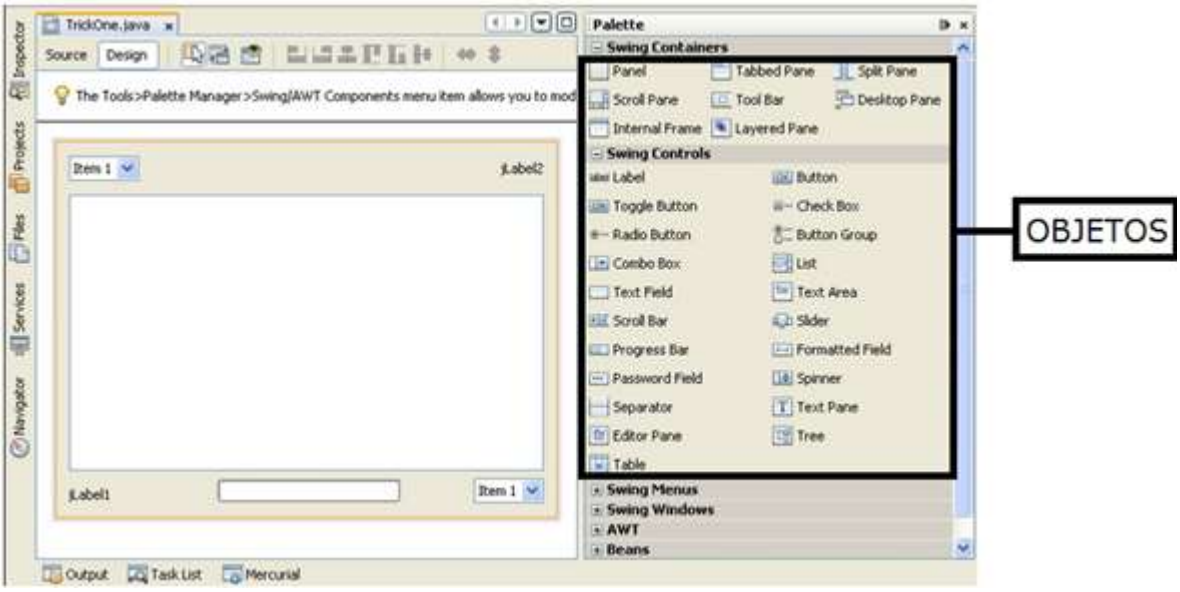
   - ✔ Peso en kilogramos:
   - ✔ Estatura en metros:

   Según sea el resultado del cálculo, mostrar un enunciado donde diga si la persona tiene:

   - ✔ Bajo peso
   - ✔ Peso normal
   - ✔ Sobrepeso
   - ✔ Obesidad grado I
   - ✔ Obesidad grado II
   - ✔ Obesidad grado III

Tomando en cuenta los IMC de la imagen siguiente:

2. **Arquitectura para considerar en toda la creación del programa:**



JUSTIFICACIÓN

Las estructuras de control, son de gran importancia dentro de la solución de problemas y/o situaciones que se presentan, siendo clave las siguientes observaciones:

1. Permiten **flexibilidad y adaptabilidad.**
2. Al hacer uso de ellas, permiten una **organización**, ya que permiten hacer una división del código de manera lógica.
3. Son **Eficientes** ya que evitan la repetición innecesaria de código lo que es de gran ayuda dentro de la resolución de problemas complejos.

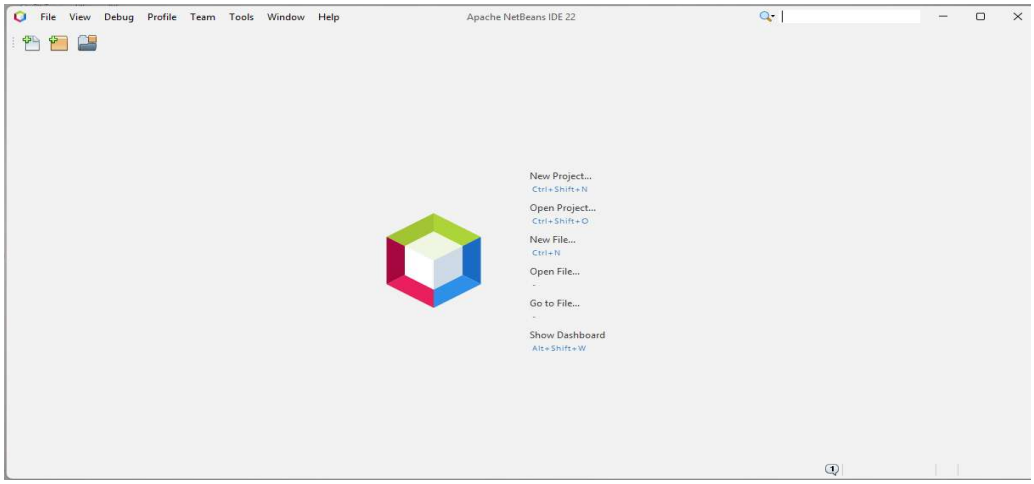
Las estructuras de control estarán controlando la ejecución de un algoritmo, a la vez que estarán permitiendo el cambio, control y modificación del orden de las instrucciones, para que el programa que estemos desarrollando sea más flexible y eficiente.

Por tanto podemos concluir que las estructuras de control con herramientas que si o si todo programador debe de utilizar, para la escritura de un código eficiente y organizado; sin embargo, es necesario tomar en cuenta lo siguiente:

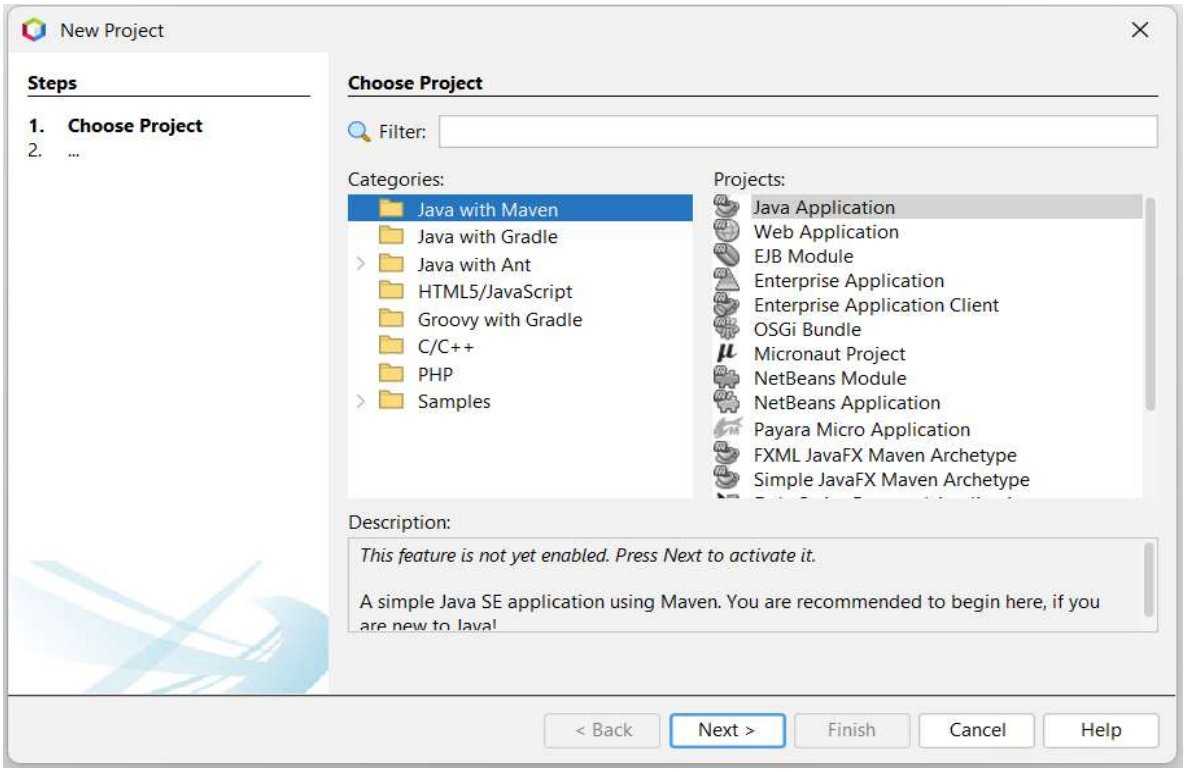
1. Evitar bucles infinitos
2. Mantener la modularidad del código
3. Priorizar la legibilidad

DESARROLLO

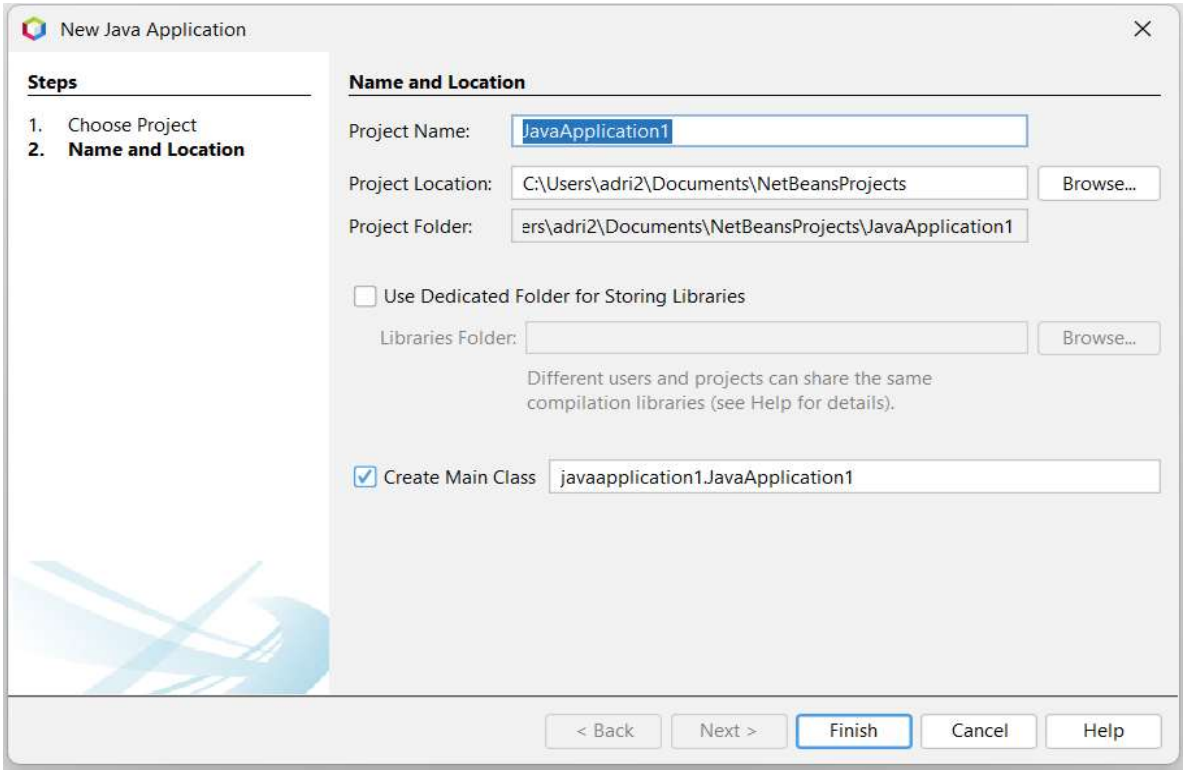
Iniciamos con la apertura de la herramienta de NetBeans



Vamos a seleccionar la opción **New Project**

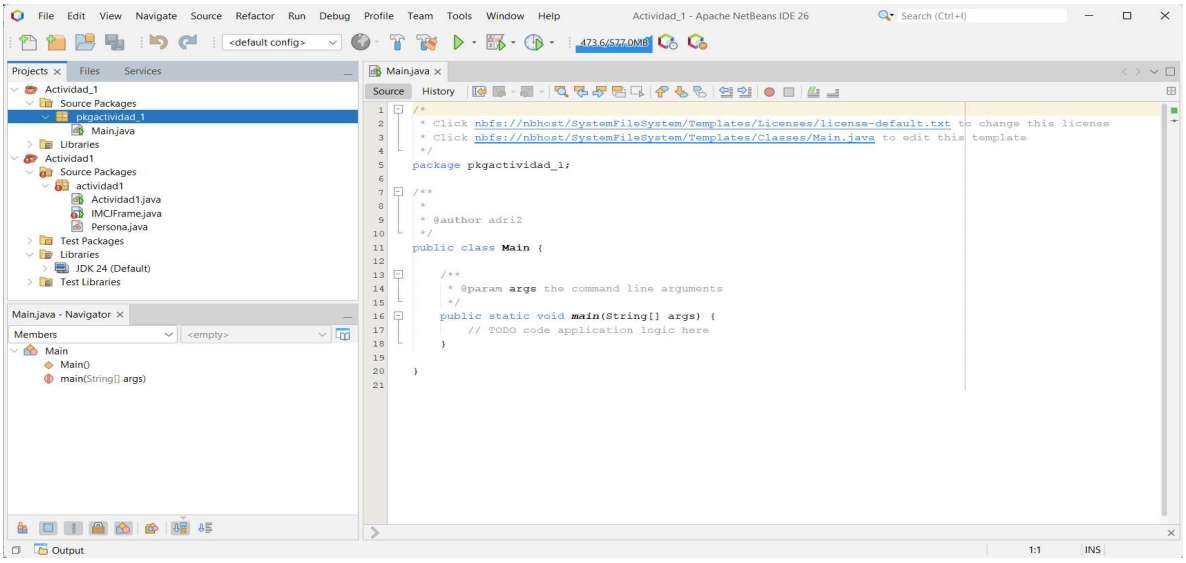


En cuanto seleccionamos la opción de New Project, estaremos visualizando esta pantalla en la cual ahora estaremos seleccionando **Java with Ant** en la columna de Categories y la opción de **Java Application** en la columna de Projects, avanzamos dando clik en el botón de **Next**



En esta ventana, estaremos definiendo nombre del proyecto y la ruta en al cual se estará guardando nuestro proyecto, para efectos de la elaboración de esta actividad, el nombre que estaremos indicando será de **Actividad_1**, finalizamos esta pantalla dando click en el botón de **Finish**.

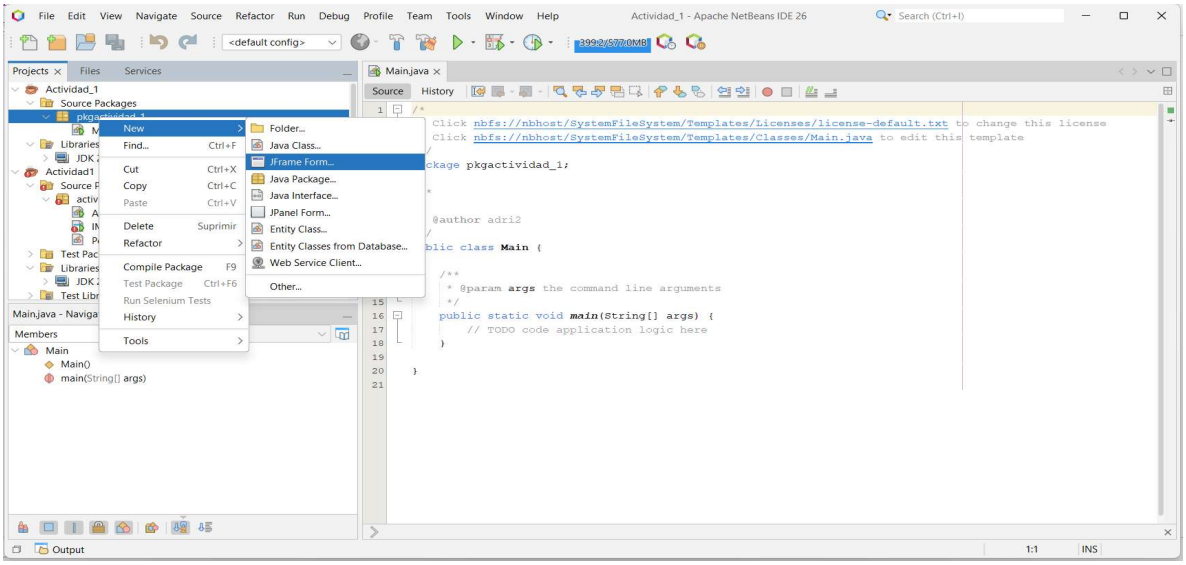
Es importante verificar que en la opción de **Create Main Class** la casilla este activada y el nombre que aparezca sea con esta sintasix: **pkgactividad_1.main**; para que una vez que se genere el proyecto en la Carpeta de **Source Packages** se este generando la el elemento **pkgactividad_1**



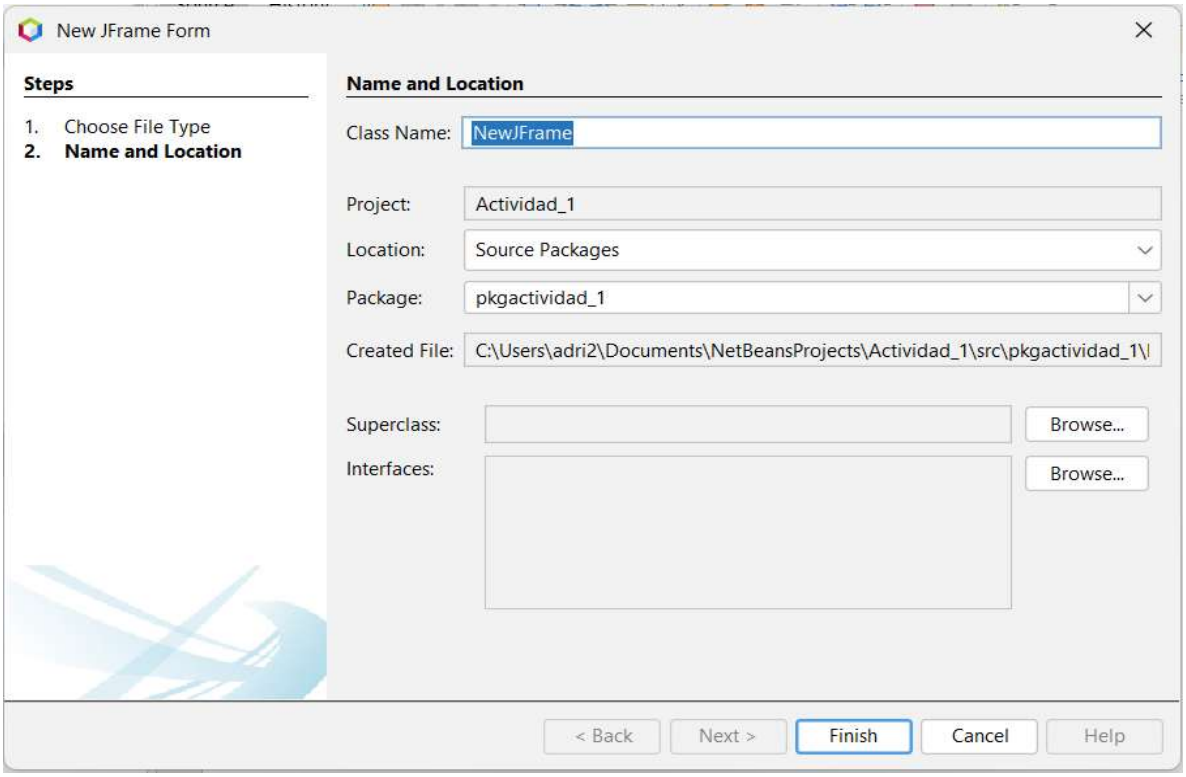
Una vez que se genera nuestro proyecto, esta es la interfaz en la cual estaremos trabajando para la realización de la actividad solicitada en la Descripción de esta actividad.

Desarrollo de Interfaz

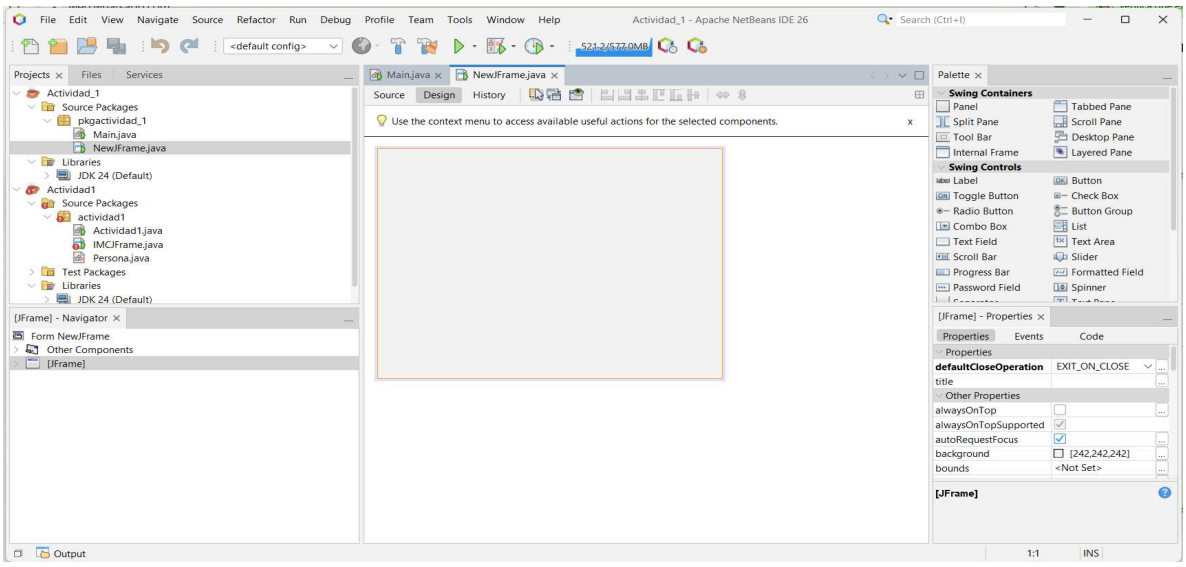
De lado izquierdo de la pantalla en la opción de **Source Packages** en donde dice **pkgactividad_1** (este nombre cambia de acuerdo al nombre del proyecto que le demos) damos click derecho seleccionamos **New** y ubicamos la opción que dice **Jframe Form**



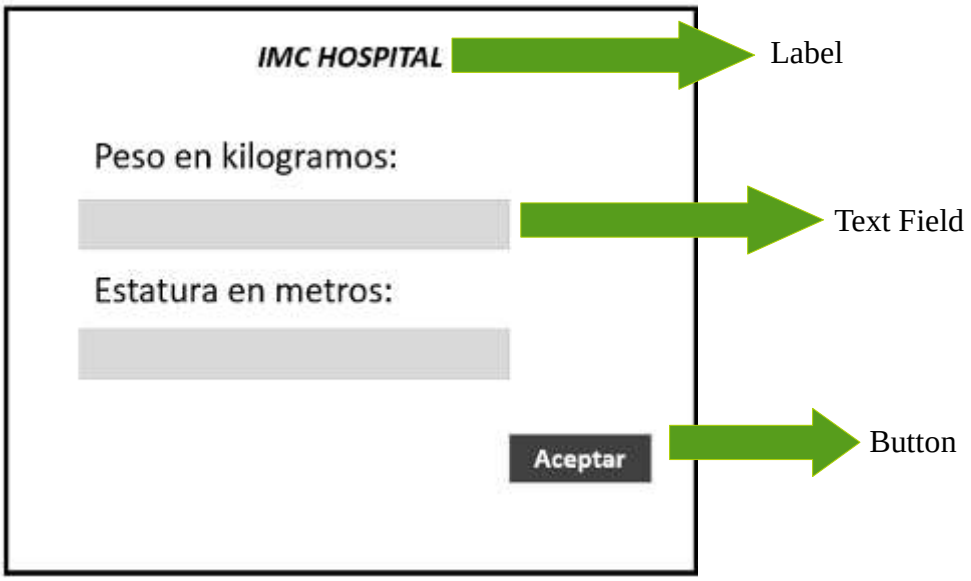
En la siguiente pantalla, solo colocamos el nombre y damos click en Finish:



Ahora, en la siguiente pantalla vamos a colocar el nombre en este caso de **IMCJFrame** y damos clik en **Finish**:



En esta área que hemos remarcado con rojo, es en dónde estaremos agregando los elementos que se están solicitando para esta actividad:

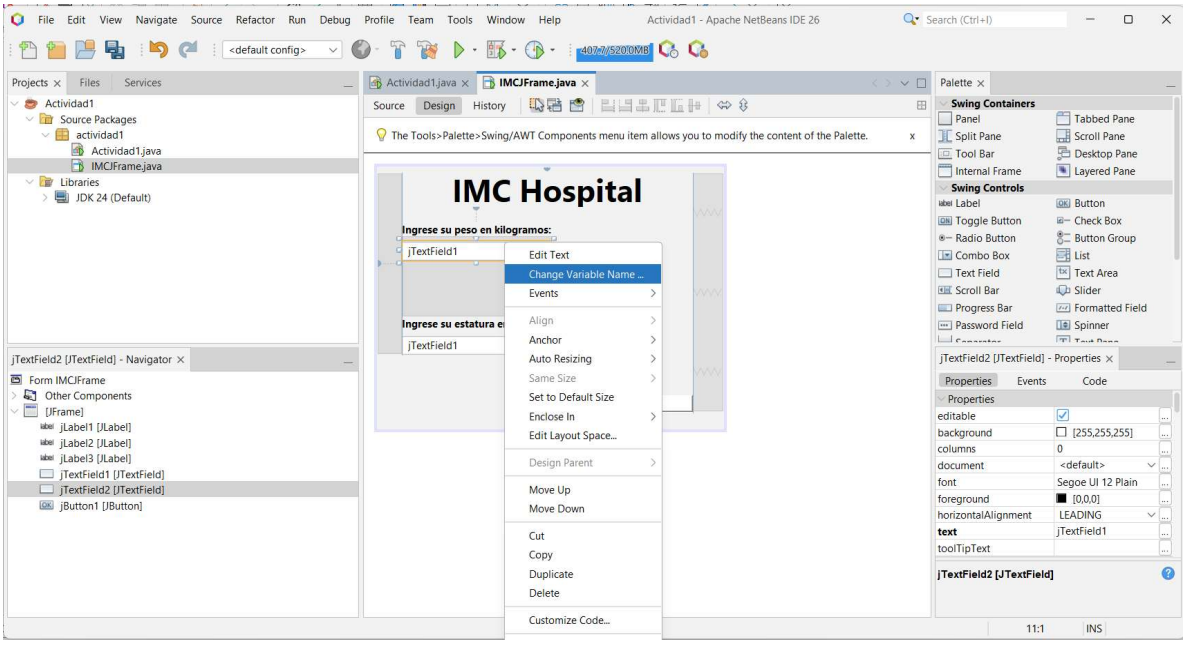


En base a esto vamos a diseñar nuestra interfaz, para lo cual estaremos haciendo uso de las herramientas que están ya predeterminadas en nuestra pantalla del lado derecho.

En esta interfaz estaremos haciendo uso de:

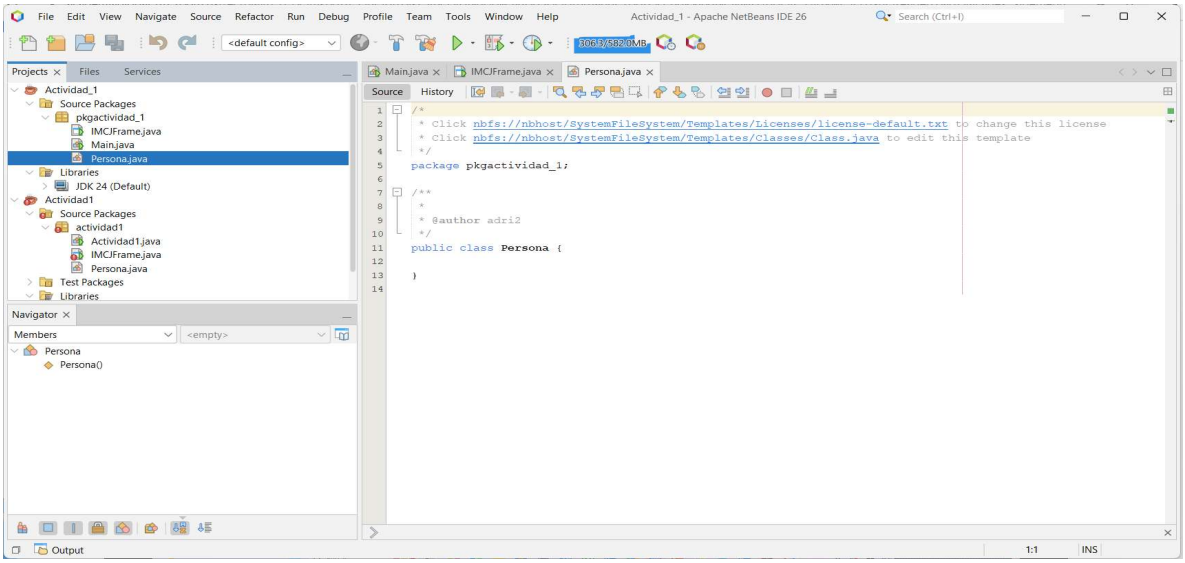
1. **Label** el cual nos permite ingresar texto en la interfaz, ya sea un título o un texto como tal, en el cual se le dan indicaciones al usuario.
2. **Text Field** en el cual se va a estar almacenando el valos que ingrese el usuario, a este elememto si es necesario darle un nombre para posteiormente poder hacer uso de esa información dentro de la programación que se requiere.
3. **Button**, el estaremos utilizando para hacer el cálculo correspondiente.

Ahora, para poder definir el nombre aquellos elementos que estarán guaradando la información que el usuario ingrese, basta con dar click derecho sobre el elemento:

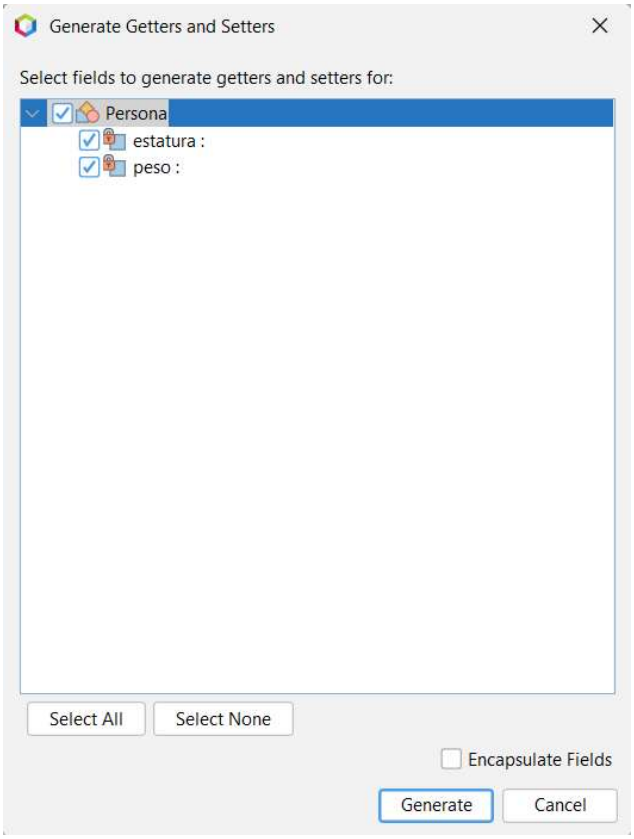


Esta es la interfaz que estará visualizando el usuario, sin embargo, aún no realiza ninguna acción, ya que es lo que estaremos programando:

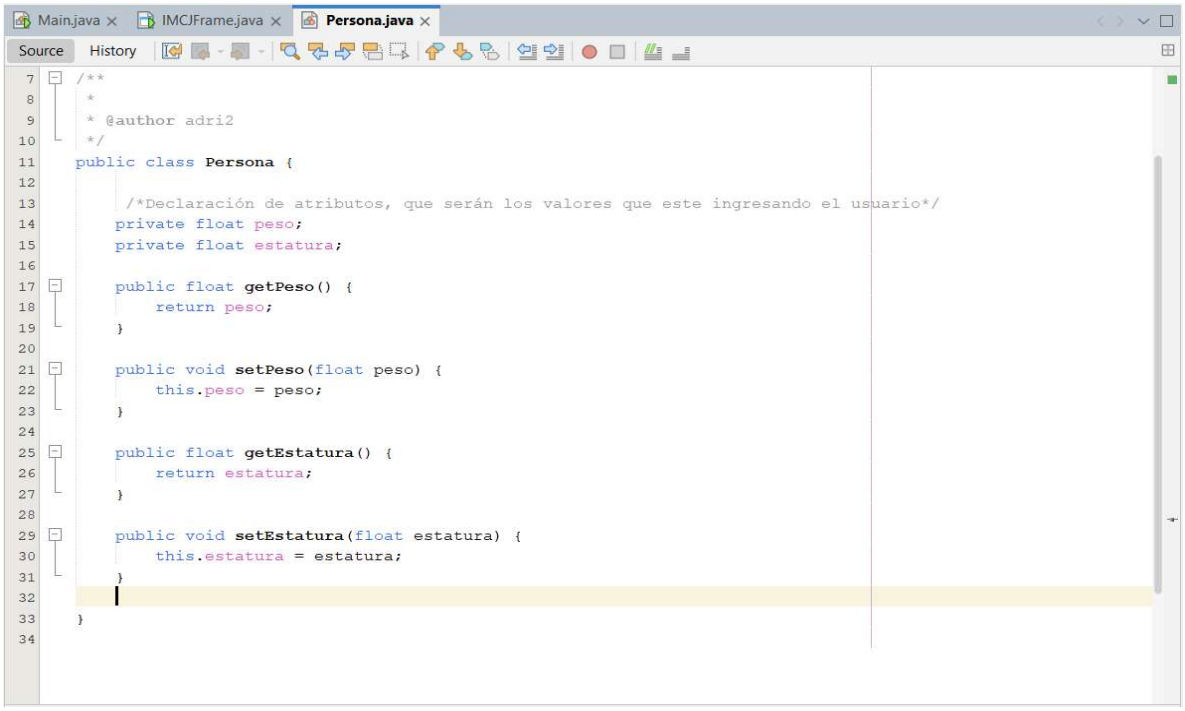
Empezamos con la programación que estará dando paso a la funcionalidad de esta interfaz, para ello v vamos a crear una clase que se llame Persona, realizamos la misma acción de dar click derecho en **pkgactividad_1**/New/Java Class/ y colocamos el nombre y damos click en Finish:



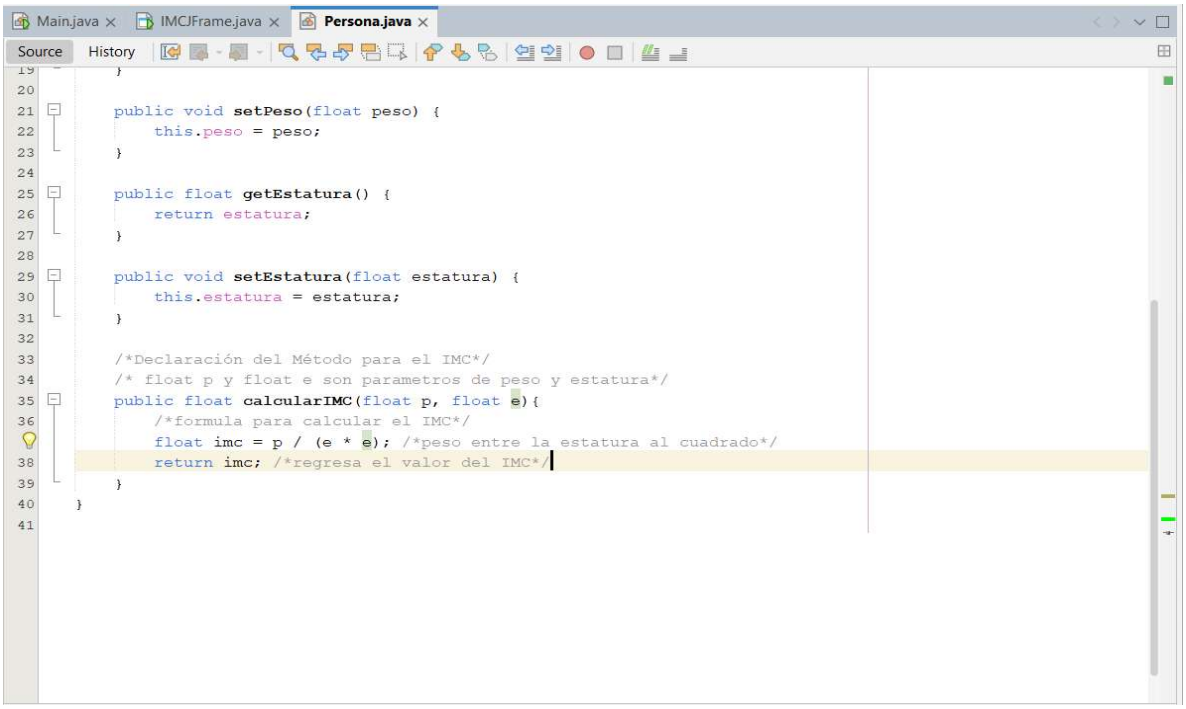
Aquí ya se genero la Clase llamada Persona, vamos a declarar los atributos de peso y estatura; ahora damos click derecho sobre el espacio de captura de código y seleccionamos la opción de **Insert Code/Getter and Setter**:



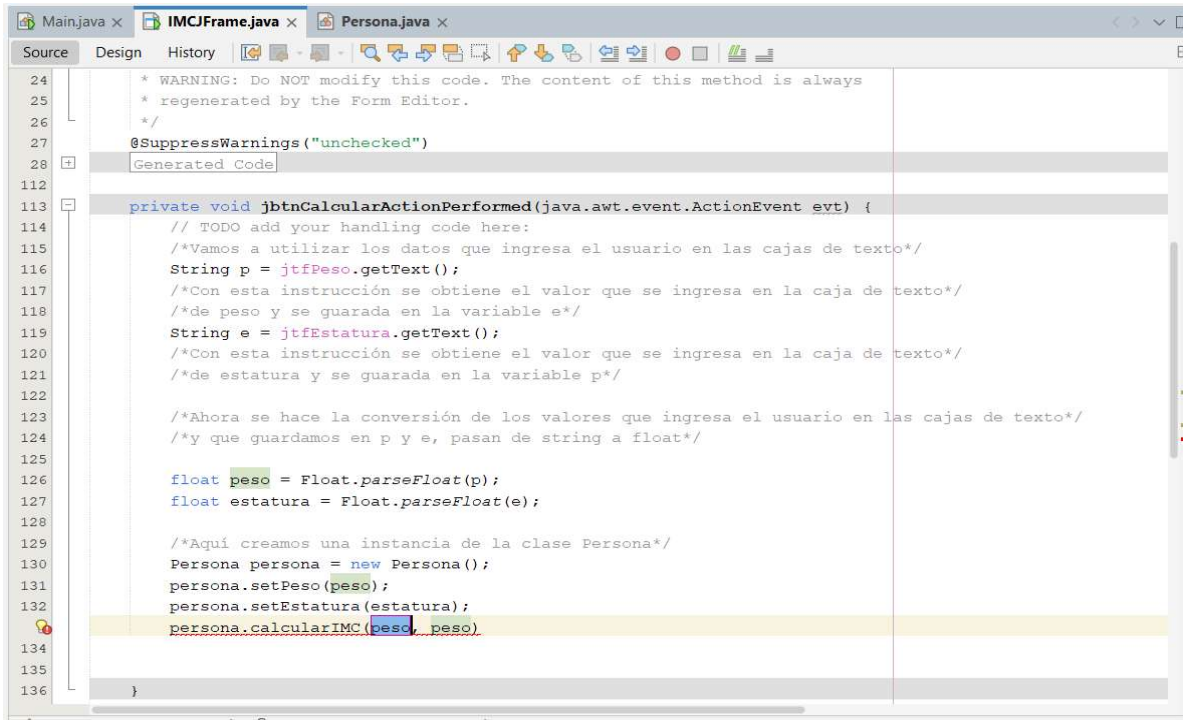
Aquí vamos a activar la casilla que dice Persona (es el nombre de nuestra clase), para generar el get y set para ambas variables de peso y estatura:



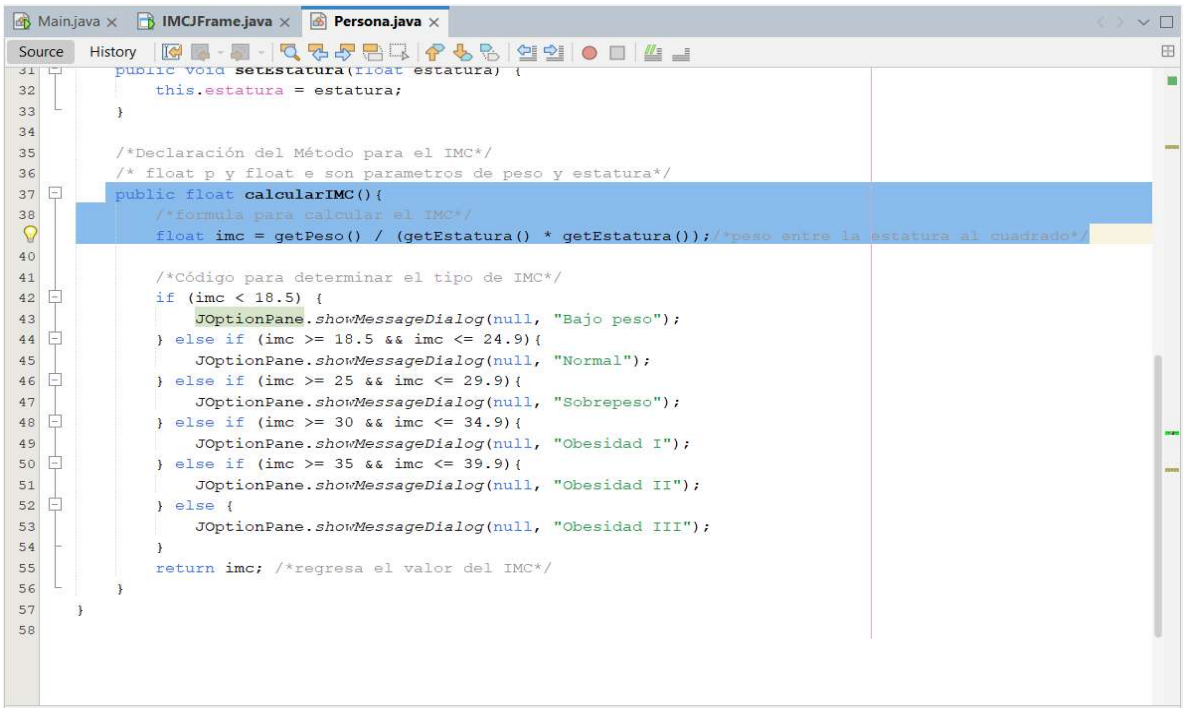
Ahora vamos hacer la declaración de un método a través del cual vamos a obtener el valor de Indice de Masa Corporal (IMC) del usuario:



Vamos a regresar a la interfaz que diseñamos, para dar doble click en el Botón y nos genere el evento:



Regresamos a la Clase Persona y realizamos lo siguiente:

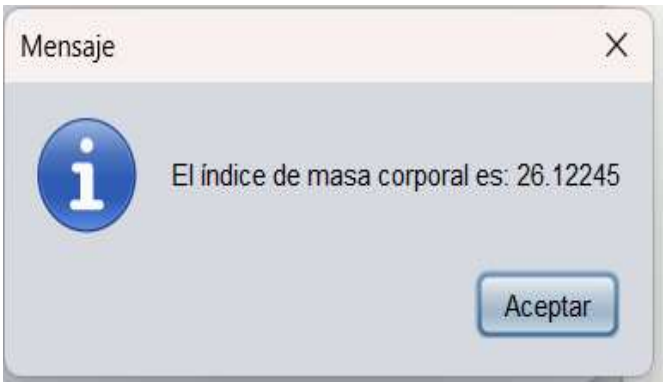


Regresamos al código del botón y realizamos lo siguiente:

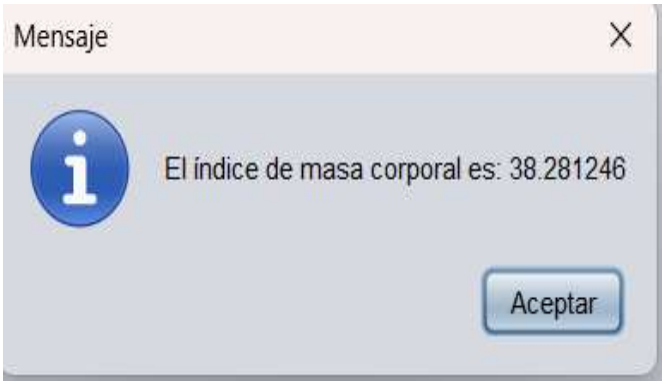
```
Main.java × | IMCJFrame.java × | Persona.java ×
Source  Design  History
116         // TODO add your handling code here:
117         /*Vamos a utilizar los datos que ingresa el usuario en las cajas de texto*/
118         String p = jtfPeso.getText();
119         /*Con esta instrucción se obtiene el valor que se ingresa en la caja de texto*/
120         /*de peso y se guarada en la variable e*/
121         String e = jtfEstatura.getText();
122         /*Con esta instrucción se obtiene el valor que se ingresa en la caja de texto*/
123         /*de estatura y se guarada en la variable p*/
124
125         /*Ahora se hace la conversión de los valores que ingresa el usuario en las cajas de texto*/
126         /*y que guardamos en p y e, pasan de string a float*/
127
128         float peso = Float.parseFloat(p);
129         float estatura = Float.parseFloat(e);
130
131         /*Aquí creamos una instancia de la clase Persona*/
132         Persona persona = new Persona();
133         persona.setPeso(peso);
134         persona.setEstatura(estatura);
135         JOptionPane.showMessageDialog(null, "El índice de masa corporal es: " + persona.calcularIMC());
136
137
138
139     }
140
141     /**
142      * @param args the command line arguments
143      */
144     public static void main(String args[]) {
145         /* Set the Nimbus look and feel */
```

Ahora vamos a ejecutar:

Al ingresar un peso de 80 kgs y una estatura de 1.75 metros, tenemos que el IMC es de 26.12 y por tanto se esta en un Sobrepeso:



Veamos otro ejemplo:



Vamos agregar estas líneas de texto dentro del Frame:

```
jtfPeso.setText("");
jtfEstatura.setText("");
```

Con esto al momento de ejecutar el formulario, volverán a estar en blanco las cajas de texto.

Se anexa el código utilizado para la Clase Persona:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package pkgactividad_1;

import javax.swing.JOptionPane;

/**
 *
 * @author adri2
 */
public class Persona {

    /*Declaración de atributos, que serán los valores que este ingresando el usuario*/
    private float peso;
    private float estatura;

    public float getPeso() {
        return peso;
    }

    public void setPeso(float peso) {
        this.peso = peso;
    }

    public float getEstatura() {
        return estatura;
    }
```

```
    public void setEstatura(float estatura) {
        this.estatura = estatura;
    }

    /*Declaración del Método para el IMC*/
    /* float p y float e son parametros de peso y estatura*/
    public float calcularIMC(){
        /*formula para calcular el IMC*/
        float imc = getPeso() / (getEstatura() * getEstatura());/*peso entre la estatura al cuadrado*/

        /*Código para determinar el tipo de IMC*/
        if (imc < 18.5) {
            JOptionPane.showMessageDialog(null, "Bajo peso");
        } else if (imc >= 18.5 && imc <= 24.9){
            JOptionPane.showMessageDialog(null, "Normal");
        } else if (imc >= 25 && imc <= 29.9){
            JOptionPane.showMessageDialog(null, "Sobrepeso");
        } else if (imc >= 30 && imc <= 34.9){
            JOptionPane.showMessageDialog(null, "Obesidad I");
        } else if (imc >= 35 && imc <= 39.9){
            JOptionPane.showMessageDialog(null, "Obesidad II");
        } else {
            JOptionPane.showMessageDialog(null, "Obesidad III");
        }
        return imc; /*regresa el valor del IMC*/
    }
}
```

Se anexa el código utilizado en el botón:

```
   private void jbtnCalcularActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
      /*Vamos a utilizar los datos que ingresa el usuario en las cajas de texto*/
      String p = jtfPeso.getText();
      /*Con esta instrucción se obtiene el valor que se ingresa en la caja de texto*/
      /*de peso y se guarada en la variable e*/
      String e = jtfEstatura.getText();
      /*Con esta instrucción se obtiene el valor que se ingresa en la caja de texto*/
      /*de estatura y se guarada en la variable p*/

      /*Ahora se hace la conversión de los valores que ingresa el usuario en las cajas de texto*/
      /*y que guardamos en p y e, pasan de string a float*/

      float peso = Float.parseFloat(p);
      float estatura = Float.parseFloat(e);

      /*Aquí creamos una instancia de la clase Persona*/
      Persona persona = new Persona();
      persona.setPeso(peso);
      persona.setEstatura(estatura);
                  JOptionPane.showMessageDialog(null, "El índice de masa corporal es: " +
persona.calcularIMC());

      jtfPeso.setText("");
      jtfEstatura.setText("");

   }
   /**
    * @param args the command line arguments
    */
```

```
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
                                                for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ReflectiveOperationException | javax.swing.UnsupportedLookAndFeelException ex) {
            logger.log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(() -> new IMCJFrame().setVisible(true));
    }
    // Variables declaration - do not modify
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JButton jbtnCalcular;
    private javax.swing.JTextField jtfEstatura;
    private javax.swing.JTextField jtfPeso;
    // End of variables declaration
}
```

CONCLUSIONES

Desde siempre, cada que buscamos solucionar un problema o construir e independientemente de que seamos o no programadores, nos plateamos una serie de pasos a seguir para dar solución a nuestra situación o problemática, y eso en términos de programación es un diagrama de flujo, lo cual es el inicio del poder llevar a cabo una programación de éxito sea cual sea el lenguaje de programación que se utilice.

Entonces una vez que ya se tiene el diagrama de flujo con la solución a la problemática, viene la parte de poder plasmar ese diagrama en términos de códigos y es ahí donde entran las estructuras de control, que nos van ayudar a plasmar un orden lógico en cuanto a la secuencia de pasos para que el programa que se este desarrollando pueda estar funcionando tal cual los requerimientos que se necesitan para la solución de la problemática que se tiene.

Dentro de este uso de la estructuras de control, también se debe de tener claro que tipo de estructura se estará utilizando, y algo importante a no olvidar, es que no necesariamente se debe de utilizar un tipo de estructura dentro de toda la codificación, sino que esto es tan flexible que se va adaptando a las necesidades de la codificación.

Se agrega dicha actividad a la plataforma de GitHub a través del siguiente link:
https://github.com/22HADRIA/Lenguajes-de-Programaci-n-IV

REFERENCIAS

¿Qué son las estructuras de control en programación?

https://keepcoding.io/blog/que-son-estructuras-de-control-en-programacion/

Estructuras de Control

https://aniei.org.mx/paginas/uam/CursoIP/curso_ip_09.html

Estructuras de Control en Programación: Guía Básica

https://futurmatica.com/estructuras-de-control-en-programacion/